Course Title:

System Analysis and Design (BIT 253)

BIT 4th semester-Tribhuvan University (TU)

Compiled by Ujjwal Rijal
rijalujjwal09@gmail.com

Unit-4

Design

Compiled by Ujjwal Rijal
rijalujjwal09@gmail.com

❑ Database Design

- The design phase focuses on the detailed implementation of system recommended in feasibility study. Emphasis is on translating performance specifications into design specifications.
- The design phase is a transition from a user – oriented document to be a document – oriented to the programmers or database personnel.
- ***Database design can be categorized as logical and physical design.***
- **In logical design**, a specification of principle features of the new system which satisfies the system's objectives are produced by the designer.
- ***Logical design*** results in the blueprint of the new system.
- **The physical design** relates to the actual input and output processes of the system. Here, the physical design follows the logical design.

❑ Process of designing Database

✓ *The database design process can be divided into the following six steps:*

➢ **Requirement Analysis**

- It is the very first step in designing a database to understand what data is to be stored, what application must be built, and what operations are most subjects to the performance requirements.

➢ **Conceptual Database Design**

- The information gathered in the requirement analysis phase is used to develop a high level description of the data to be stored in the database, along with the constraints known to be hold over this data.

➢ **Logical Database Design**

- It is used to convert the conceptual database design into a database schema in the data model of the chosen DBMS. Normally, we will consider a relational DBMS (RDBMS).

Contd. ….

➢ Schema Refinement

- This step analyses the collection of relations in our relational database schema to identify the potential problems, and refine it.

➢ Physical Database Design

- This step involves building indexes on some table and clustering some tables or it may involve re – design of parts of database schema obtained from the earlier steps.

➢ Application and Security Design

- Any software project that involves a DBMS must consider the aspects of the application that goes beyond the database itself. We must consider the role of each entity in every processes.
- A DBMS provides several mechanisms to assist in this step.

❑ Relational Database Model

- A relation is a named, two dimensional table of data. Each relational consists of a set of named column and an arbitrary (any / unlimited) number of unnamed rows.

Table: Employee relation with sample data

Employee_ID	Name	Salary (in RS.)
1	Divya	25, 000
2	Kritika	30, 000
3	Prapti	35, 000
4	Avni	40, 000
5	Bibisa	45, 000

Contd. ….

- The relational database model represents the data in the form of related table or relation.
- The above table shows relation named as "EMPLOYEE". This relation consists of the attributes: "Employee_ID", "Name", and "Salary".
- Here, the structure of relation with a short hand notation is as follows:

EMPLOYEE (Employee_ID, Name, Salary);

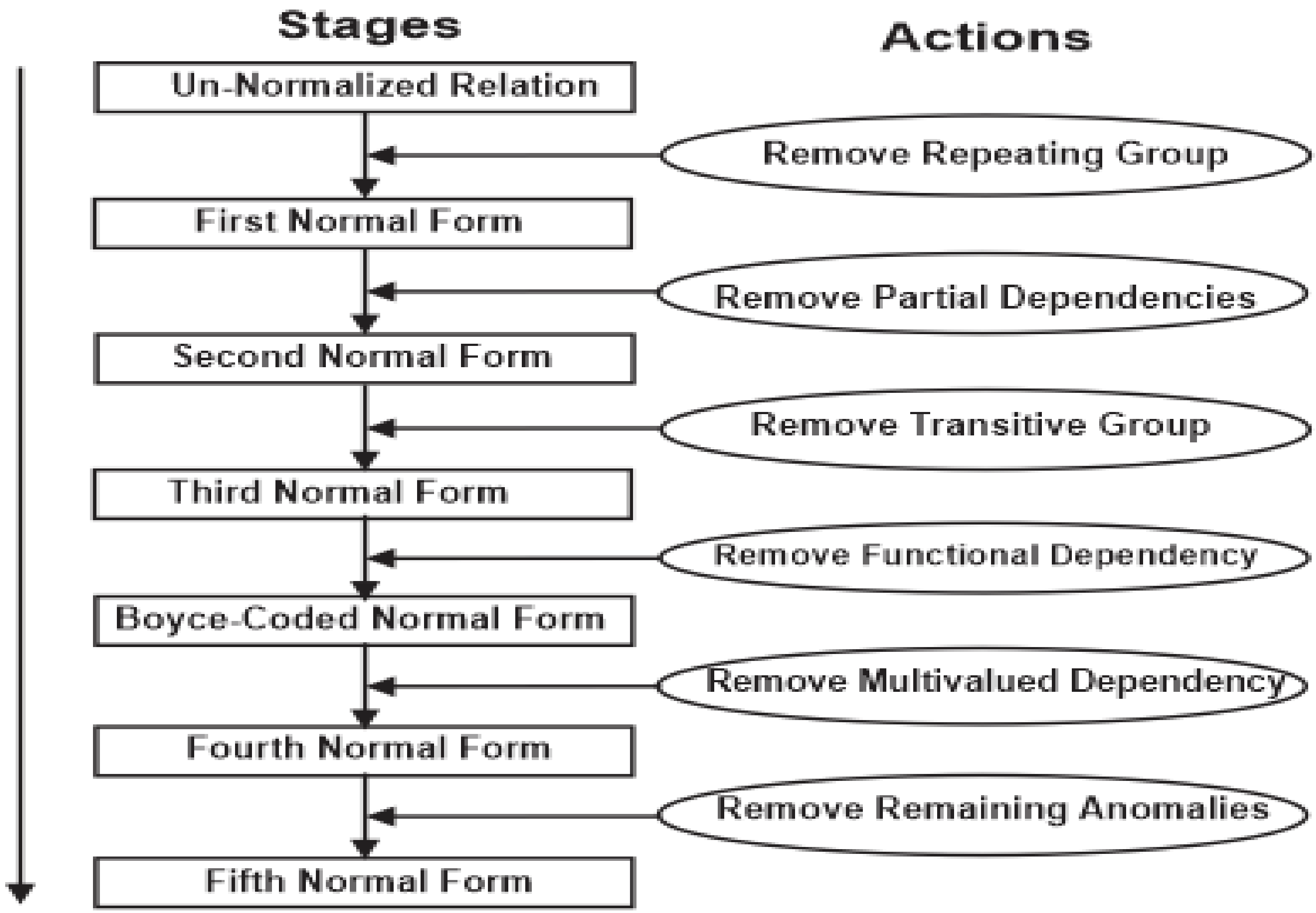
❑ Normalization

- Normalization is the process of creating table design by assigning fields or attributes to each table in the database. A table design specifies the fields and identifies the primary key in a particular table or file.
- Actually, the process of converting complex data structure into simple, stable data structure is known as normalization.
- In other words, the process of decomposing unsatisfactory larger tables or relations by breaking of their attributes into smaller relations or tables is known as normalization.
- The well structured relation contains minimal redundancy, and allows insertion, modification, and deletion without errors or inconsistencies, and maintains data integrity.

❑ Advantages of Normalization

- Minimize redundancy from table or relation.
- Manage insertion, modification and deletion of anomalies or inconsistencies.
- Helps to maintain data consistencies in the database.
- Puts the data into the form that is more able to accurately accommodate or sustain the change.
- Facilitates the enforcement of data constraints.
- Avoids unnecessary coding, etc.

❑ Steps/ Phases/ Types of Normalization



Stages

Actions

Un-Normalized Relation

Remove Repeating Group

First Normal Form

Remove Partial Dependencies

Second Normal Form

Remove Transitive Group

Third Normal Form

Remove Functional Dependency

Boyce-Coded Normal Form

Remove Multivalued Dependency

Fourth Normal Form

Remove Remaining Anomalies

Fifth Normal Form

Contd. ….

Un – Normalized Form (UNF)

- A table design that contains a repeating group is called UNF design. The standard notation method for representing an UNF design is to enclose the repeating group of its within a second set of parenthesis.
- **An example of an UNF table is as follows:**

NAME(field1, field2, field3 (repeating field1, repeating field2));

First Normal Form (1NF)

- A relation is said to be in 1NF if and only if all the domains of the relation contains only the atomic (or indivisible or simple) values.
- Moreover, a relation is in 1NF if it does not have the multi – valued attributes, composite attributes, and their combination.

Contd. ….

- Let us take an UNF relation containing multi – valued attributes:

Student_Table

S.N.	Name	Address	Phone_No.
1	Shiva	Brt – 1	9812777809, 9842500021
2	Parbati	Brt – 2	9816949887
3	Radha	Brt – 3	9815599837
4	Krishna	Brt – 4	9816949847

- Now, the 1NF of the above UNF table is as follows:

Contd. ….

Student_Table

S.N.	Name	Address	Phone_No._1	Phone_No._2
1	Shiva	Brt – 1	9812777809	9842500021
2	Parbati	Brt – 2	9816949887	Null
3	Radha	Brt – 3	9815599837	Null
4	Krishna	Brt – 4	9816949847	Null

Second Normal Form (2NF)

- A relation is said to be in 2NF if and only if it is already in 1NF. Every non – prime attributes are fully dependent on the primary key of the relation.
- Inversely, if a table has some attributes which are partially dependent on the primary key of that table, then it is not in 2NF.

Contd. ….

Example: Consider the following relation which is not in 2NF:

Employee_Department_Table

Employee_ID	Employee_Name	Employee_Salary	Dept_No.	Dept_Name
1	Shiva	40, 000	D_1	BSc. CSIT
2	Parbati	30, 000	D_2	BIT
3	Radha	20, 0000	D_3	BCA
4	Krishna	10, 000	D_4	BIM

- Now, the above table in 2NF looks as follows:

Contd. ….

Employee_ID	Employee_Name	Employee_Salary	Dept_No.
1	Shiva	40, 000	D_1
2	Parbati	30, 000	D_2
3	Radha	20, 0000	D_3
4	Krishna	10, 000	D_4

Employee_ID	Employee_Name	Employee_Salary	Dept_Name
1	Shiva	40, 000	BSc. CSIT
2	Parbati	30, 000	BIT
3	Radha	20, 0000	BCA
4	Krishna	10, 000	BIM

Contd. ….

- **Third Normal Form (3NF)**
- **A relation is said to be in 3NF if and only if:**
- ✓ **It is always in 2NF.**
- ✓ **Every non – prime attributes is non – transitively dependent on the primary key.**
- ***Speaking inversely, if a table contains transitive dependency, then it is not in 3NF and the table must be partitioned to bring it into 3NF.***

For example: if a → b and b → c, then a → c.

i.e. if (a → b → c) form occurs in a table, then it is not in 3NF.

Contd. ….

Example: Consider the following relation which is not in 3NF:

Student_Table

S_ID	S_Name	Age	Gender	Hostel_Name
1	Shiva	21	M	Sunrise
2	Krishna	22	M	Sunrise
3	Radha	23	F	Suryajyoti
4	Parbati	24	F	Suryajyoti

- Here, we have:

S_ID → S_Name

S_ID → Age

S_ID → Gender

S_ID → Hostel_Name

Gender → Hostel_Name

Contd. ….

- This last dependency Gender → Hostel_Name was not originally specified but to have derived it. This derived dependency is called a transitive dependency.
- In such a case, the relation should be broken into relations to make it in 3NF as follows:

S_ID	S_Name	Age	Gender
1	Shiva	21	M
2	Krishna	22	M
3	Radha	23	F
4	Parbati	24	F

Gender	Hostel_Name
M	Sunrise
F	Suryajyoti

❑ Transforming ER – Diagrams into Relations

- Transforming ER – Diagrams into normalized relations and then merging all the relations into one final set of relations can be accomplished within the following four steps:

I. Represent Entities

- Each entity type in the ERD becomes a relation. The identifier of the entity types becomes the primary key of the relation, and other attributes become non – prime key attributes of the relation.

II. Represent Relationships

- Each relationship in an ERD must be represented in the relational database design. How we represent a relationship depends on its nature. Relationships may be 1:1, 1:N, M:N, etc.

III. Normalize the Relations

- The relations created in steps 1 and 2 may have unnecessary redundancies. So, we need to normalize these relations to make them well – structured.

IV. Merge the Relations

- So far in database design, we have created various relations from both a bottom – up normalization of user views and from transforming one or more ERDs into set of relations. Across these different set of relations, there may be redundant relations that must be merged and re – normalized to remove the redundancies.

❑ Merging the Relations

- As a part of the logical database design, normalized relations likely have been created from a number of separated ERD and various user interfaces.
- Some of the relations may be redundant, so to merge the relations, we must define the objectives of removing the redundancies.

Example of merging the relations:

- Suppose that modelling a user interface or transforming an ERD results in the following 3NF relation:

  EMPLOYEE (Emp_ID, Name, Address, Phone_No.);

- Modelling a second user interface might result in the following relation:

  EMPLOYEE (Emp_ID, Name, Address, Job, Number_of_Year);

Contd. ….

- Because these two relations have the same primary key (Emp_ID) and describe the same entity, they should be merged into one relation.
- **The result of merging the relation is as follows:**

  EMPLOYEE (Emp_ID, Name, Address, Phone_No., Job, Number_of_year);
- **Note that**, attribute that appears in both the relations (such as name in this example) appears only once in the merged relation.

❑ Physical File and Database Design

- Designing physical files and databases requires certain information that should have been collected and produced during prior SDLC phases.
- ***This information includes the following:***

✓ Normalized relations, including volume estimates.

✓ Definitions of each attribute.

✓ Descriptions of where and when data are used: entered, retrieved, deleted, and updated (including frequencies).

✓ Expectations or requirements for response time and data integrity.

✓ Descriptions of the technologies used for implementing the files and database so that the range of required decisions and choices for each is known.

- Thus, we begin the physical design phase by addressing the **design of physical fields** and the **design of physical tables** for each attribute in a logical data model.

❑ Designing Fields

- A field is the smallest unit of application data recognized by system software, such as programming language or DBMS.
- An attribute from a logical database model may be represented by the several fields. For example: a student name attribute in a normalized student relation might be represented as three fields: last name, first name, and middle name.
- The basic decisions we must take in specifying each field concern the type of data (or storage type) used to represent the filed and data integrity controls for the field.
- **Calculated Field:** A field that can be derived from other database fields is called a calculated field (**or a computed field or a derived field**). **For example**, an invoice may include a total due field, which represents the sum of the amount due on each item on the invoice.

❑ Designing Physical Tables

- A relational database is a set of related tables, related by the foreign keys referencing primary keys.
- In logical database design, we group into a relation those attributes that concern some unifying, normalized business concept, such as customer, product, or employee.
- In contrast, a physical table is a named set of rows and columns that specifies the fields in each row of the table.
- A physical table may or may not correspond to one relation, whereas normalized relations possess properties of well – structured relations.
- **The design of a physical table has two goals, different from those of normalization: efficient use of secondary storage, and data processing speed.**

❑ Designing Forms and Reports

❖ Form

- A business document that contains some pre – defined data, and may include some areas where additional data are to be filled in is called a form.
- An instance of a form is typically based on one database record.

❖ Report

- A business document that contains only pre – defined data, and is a passive document used solely for reading or viewing purposes is called a report.
- A report typically contains data from many unrelated records or the transactions.

❑ Process of Designing Forms and Reports

- The system inputs and outputs are produced at the end of the analysis phase. Precise appearance was not defined during this phase. Forms and reports are integrally related to DFDs and ERDs.
- The process of designing forms and reports is a user – focused activity that typically follows a prototyping approach.
- It starts with collecting the initial requirements, and then structuring and refining those information into an initial prototype. Structuring and refining requirements are completed independently of the users.
- After reviewing the prototype or evaluating the prototype, users may accept the design or request for the changes to be made.
- If the changes are needed, we will repeat the construction – evaluation – refinement cycle until the design is accepted. The cycle might go through the multiple iterations.

Contd. ….

- ***Here, process of designing forms and reports arises the requirements determination, such as :***

✓ Who will use the form or report?

✓ What is the purpose of the form and report?

✓ When is the form and report needed or used?

✓ When does the form or report need to he delivered and used?

✓ How may people need to use or view the form or report?, etc.

- ***The deliverables and outcomes from the process of designing forms and reports are the design specifications, and they are as follows:***

➢ **Narrative Overview**

- It contains a general overview of the characteristics of the target users, tasks, system, and environmental factors in which the form or report will be used.

Contd. ….

Sample Design

- In this section, a sample design of the form is drawn. The design may be hand – drawn using a coding sheet, although in most instances, it is developed using CASE or standard development tools.

Testing and usability assessment

- It provides all the testing and usability assessment information. Procedures for assessing designs are described in this section.

❑ Formatting Forms and Reports

- *The process or guidelines of formatting forms and reports are:*

✓ **Meaningful Title**

- Clear and specific titles describing the content and use of forms and reports.
- Revision data, current data, valid date, etc.

✓ **Meaningful Information**

- Only needed information should be displayed and it should be provided in usable format without modifications.

✓ **Balance the layout**

- Information should be balanced on the screen or page, and sufficient spacing and margins should be used.

✓ **Design an easy navigation system**

- Clearly show how to move forward and backward.
- Clearly show where we are, and notify user when on the last page of a multiple sequence.

Contd. ….

- ✓ **Highlighting Information**
- Notifying users of errors in data entry or processing.
- Providing warnings to users regarding possible problems such as unusual data values or an unavailable device.
- Methods of highlighting includes: blinking and audible tones, colour differences, size differences, font differences, underlining, all capital letters, etc.
- ✓ **Displaying Text**
- In business related system, textual output is becoming increasingly important as text – based applications such as email, bulletin, and information services are more widely used, etc.

Contd. ….

- ✓ **Color vs. No Color**
- Color is a powerful tool for the designer in influencing the usability of an information system.
- When applied appropriately, color provides many potential benefits to forms and reports, such as draws attention to warnings, strikes the eyes, emphasizes the logical organization of information.
- Problems from using color are such as color – blindness, resolution may degrade with different displays, printing or conversion to other media may not easily translate the information, etc.

❑ Types of Business Report

✓ **Scheduled Report**

- Reports produced at pre – defined intervals – daily, weekly, or monthly to support the routine information needs of an organization are called scheduled reports.

✓ **Key – Indication Report**

- Reports that provide a summary of critical information in a recurring basis are called key – indication reports.

✓ **Exception Report**

- Reports that highlight data which are out of the normal operating range are called exception reports.

✓ **Drill – Down Report**

- Reports that provide details behind the summary values on a key indicator or exception reports are called drill – down reports.

✓ **Ad – hoc Report**

- These types of reports contain unplanned information requests in which information is gathered to support a non – routine decision.

❑ Assessing Usability

- There are many factors to consider when we design forms and reports. The objective for designing forms and reports, and human – computer interactions (HCI) is assessing usability.
- *Usability typically refers to the following three characteristics:*

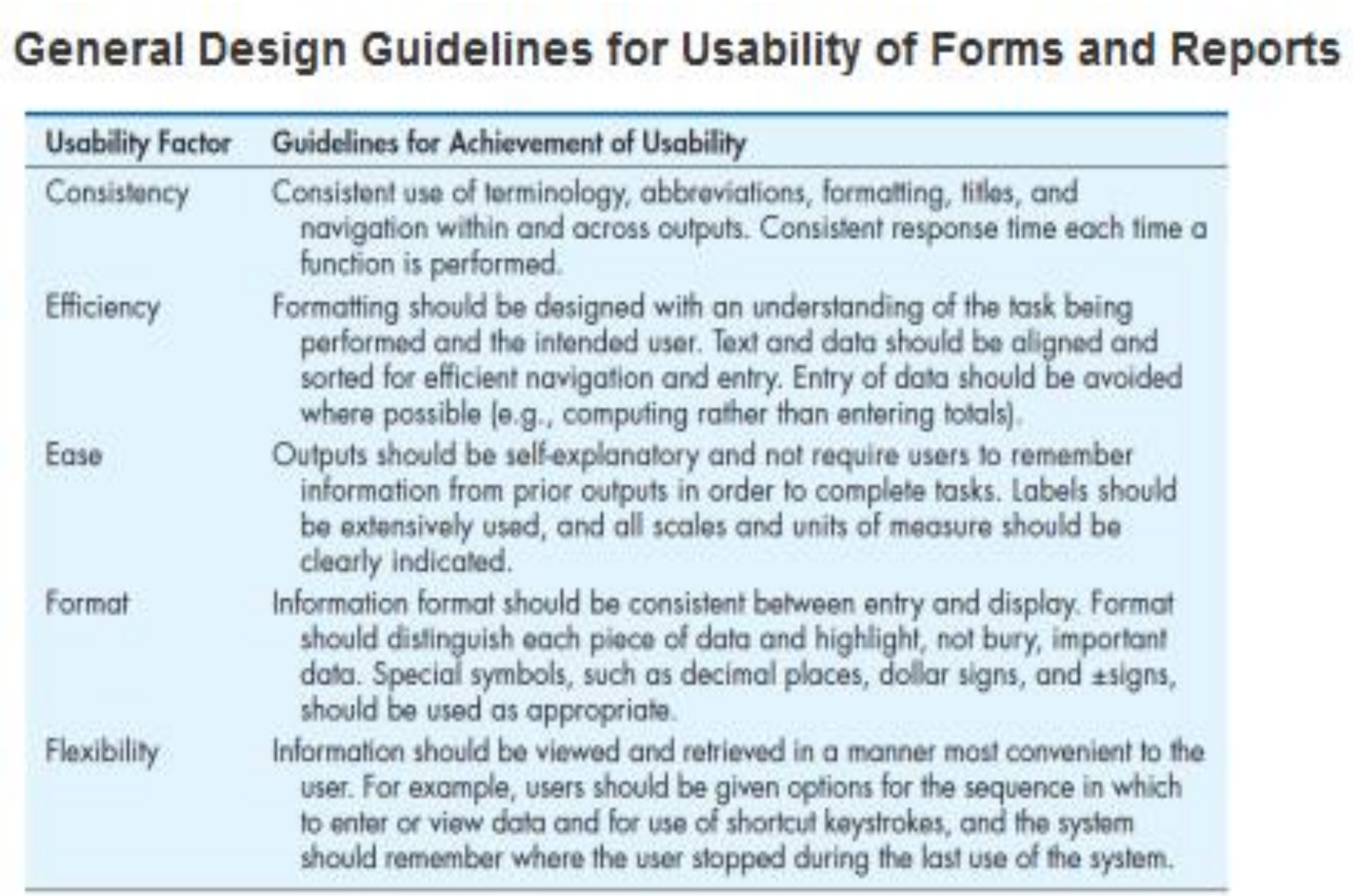
✓ **Speed:** Can we complete a task efficiently?

✓ **Accuracy:** Does the system provide what we expect?

✓ **Satisfaction:** Do we like using the system?

- In other words, usability means that our designs should assist, not hinder, user performance.
- Thus, usability refers to an overall evaluation of how a system performs in supporting a particular user for a particular task.

Contd. ….



General Design Guidelines for Usability of Forms and Reports

Usability Factor	Guidelines for Achievement of Usability
Consistency	Consistent use of terminology, abbreviations, formatting, titles, and navigation within and across outputs. Consistent response time each time a function is performed.
Efficiency	Formatting should be designed with an understanding of the task being performed and the intended user. Text and data should be aligned and sorted for efficient navigation and entry. Entry of data should be avoided where possible (e.g., computing rather than entering totals).
Ease	Outputs should be self-explanatory and not require users to remember information from prior outputs in order to complete tasks. Labels should be extensively used, and all scales and units of measure should be clearly indicated.
Format	Information format should be consistent between entry and display. Format should distinguish each piece of data and highlight, not bury, important data. Special symbols, such as decimal places, dollar signs, and ±signs, should be used as appropriate.
Flexibility	Information should be viewed and retrieved in a manner most convenient to the user. For example, users should be given options for the sequence in which to enter or view data and for use of shortcut keystrokes, and the system should remember where the user stopped during the last use of the system.

❑ Designing Interfaces and Dialogues

- The process of designing interfaces and dialogues is similar to the process of designing forms and reports.
- The process of designing interfaces and dialogues is a user – focused activity. This means that we follow the prototyping methodologies of iteratively collected information, constructing a prototype, assessing usability, and making refinements.
- To design usable interfaces and dialogues, we must answer the same who, what, when, where, and how questions used to guide the design of forms and reports.
- Thus, this process of designing interfaces and dialogues is parallel to that of designing forms and reports.
- ***The deliverables and outcomes of designing interfaces and dialogues follow the following three sections:***

✓ ***Narrative overview, Sample design, and Testing and usability assessment.***

❑ Interaction Methods and Devices

- An interface is a method by which users interact with the computer information system.
- The **human – computer interface (HCI)** defines the ways in which users interact with an information system.

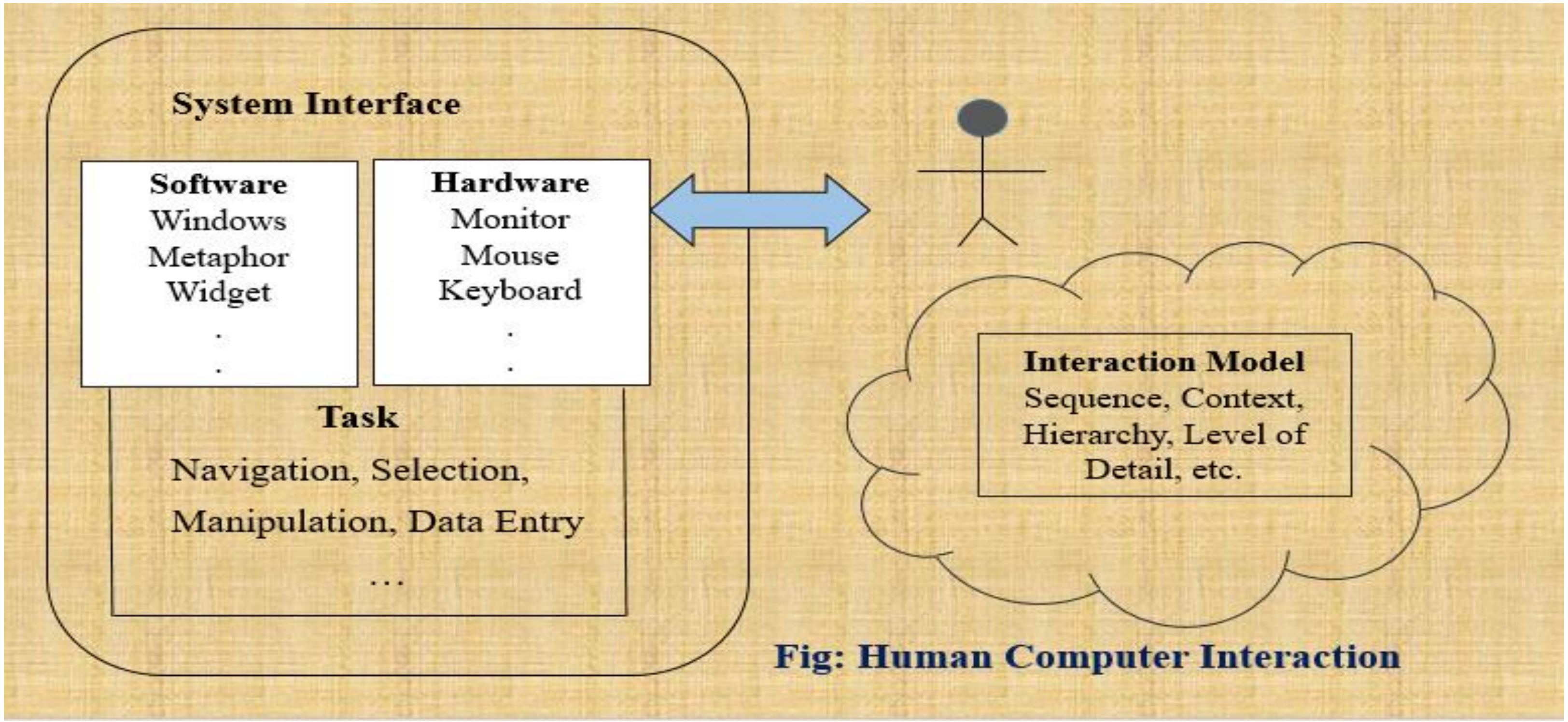


Fig: Human Computer Interaction

Contd. ….

- *There are various methods of interaction and some major of them are discussed below:*

Menu Interaction

- In this method, a user selects a command from the list of possible menu. It is often the case that another screen object is selected at the same time and the command operates on that objects. For example, to delete a file, user selects the file then selects the delete command.

Command Language Interaction

- In this method, the user issues a special command and associated parameters to instruct the system what to do. For example, to delete a file, the user issues a delete command with the filename as a parameter.

Form Interaction

- Here a user fills in the fields of a form. Some fields may have associated menus and the form may have action buttons that when pressed cause some action to be initiated. This is simple data entry and easy to learn but it takes up a lot of space in the screen.

Contd. ….

Object – based Interaction

- Here, symbols are used to represent commands or functions. The most common method for implementing object – based interaction is making the use of icons. Icons are the graphical symbols that represent the specific function within a system.

Natural Language Interaction

- In this method, the user issues a command in natural language. It is accessible to causal users and requires more typing. Natural language understanding systems are unreliable. For example, WWW information retrieval systems.

Direct Manipulation Interaction

- In this method, the user interacts directly with objects on the screen. It is fast and intuitive interaction, and easy to learn. But, it may be hard to implement, only suitable where there is a visual metaphor for the tasks and the objects.

❑ Designing Interfaces and Dialogues in Graphical Environments

- Graphical User Interface (GUI) environments have become the standard method for human – computer interaction (HCI).
- Although all of the interface and the dialogue design guidelines presented previously apply to designing GUI, additional issues that are unique to these environments must be considered.
- ***Here, some of them are discussed below:***

➢ **Graphical interface design issues**

- When designing HUI for an operating environment such as MS Windows, numerous factors must be considered.
- Some factors are common to all GUI environments where others are specific to single environment.
- ***Here, the effective GUI designer would play the role to design an interface and dialogue effectively as follows:***

✓ The first step should be an obvious one which says that become an expert user of the GUI environment.

✓ The second step focuses on understanding the available resources and how they can be used.

Contd. ….

➢ Dialogue design issues

- When designing a dialogue, our goal is to establish the sequence of displays (full screen) that users will encounter when working with the system.
- The dialogue designing process may face the issues like what, where, when, how, etc.
- Within many GUI environments, encountering process can be a bit more challenging due to the GUI's ability to suspend the activities and switch the another application or task.
- Thus, these types of issues must be resolved to obtain good dialogue designing in the graphical environments.